Virginia Tech Grand Challenge Team

DARPA Grand Challenge 2005

Dr Charles Reinholtz
creinhol@vt.edu

Brett Leedy
grandchallenge@vt.edu

Joseph Putney
jputney@vt.edu

J. Michael Webster
jowebste@vt.edu

Cheryl Bauman
cbauman@vt.edu

David Anderson
daveanderson@vt.edu

Steven Cacciola
scacciol@vt.edu

Raymond Wilhelm
raywilhe@vt.edu

1. Vehicle Description

1.1. Describe the vehicle. If it is based on a commercially available platform, provide the year, make, and model. If it uses a custom-built chassis or body, describe the major characteristics. If appropriate, please provide a rationale for the choice of this vehicle for the DGC.

The Virginia Tech base platform is an Ingersoll-Rand Club Car XRT-1500. Cliff is a pre-production prototype built in 2004. Virginia Tech has expanded the fuel capacity for longer operation and added a reinforced, expanded roll cage to accommodate and protect the sensitive electronics onboard. This vehicle suits our application well with an 11.5' turning radius, top speed of 25mph, and high (>800lb) cargo capacity.

1.2. Describe any unique vehicle drive-train or suspension modifications made for the DGC including fuel-cells or other unique power sources.

Cliff has no modifications to the stock suspension or drive train.

2. Autonomous Operations

2.1. Processing

2.1.1. Describe the computing systems (hardware and software) including processor selection, complexity considerations, software implementation and anticipated reliability.

Computational power on Cliff is distributed across three National Instruments PXI-8176 controllers. Using PCI electrical-bus features, PXIs are a high-performance and low-cost method for measurement and autonomous control. These computers range in speed from 1.26 to 2.4 gigahertz, and can withstand shocks of up to 30 g's. With this fairly high resistance to shock, the PXI chassis can be rigidly mounted inside the electrical box without fear of damage. These platforms excel in interfacing with sensors. Using the embedded control feature of the PXI eliminates the need for external computers for vehicle control.

Windows XP runs on all three computers. While this can create problems due to the inherent instability of Windows, it allows all team members to interact with Cliff. After removing all unnecessary features of the Windows operating system, it provides a relatively stable and familiar, easy to use interface to all team members.

The software on Cliff was created using National Instruments' Labview 7.1. This program allows team members with knowledge of control systems but little programming experience to program the vehicles behavior. Certain parts of the programs are written in C; however, these pieces are converted into files that are later used by the larger Labview code. Another large benefit of using Labview is the ease of creating vehicle interfaces within the programs. Any team member can easily create an interface that monitors all vehicle action during autonomous operation. This allows for quick and easy debugging to any problems that appear during testing.

2.1.2. Provide a functional block diagram of the processing architecture that describes how the sensing, navigation and actuation are coupled to the processing element(s) to enable autonomous operation. Show the network architecture and discuss the challenges faced in realization of the system.

Cliff's sensing, navigation, and actuation are controlled with three computers: Motion Control, Stereo Vision, and INS/Path Planning. Using the INS/Path Planning computer, autonomous operation begins with the vehicle determining its current position and locating its destination.

The vehicle also evaluates its surroundings using LIDAR and stereo vision. The LIDAR uses a laser to scan the area in front of the vehicle and detect obstacles. With each scan, it returns the distance to all of the obstacles it sees in the vehicles path. The stereo vision camera is also monitoring what is happening in front of the vehicle. The vision is used to determine whether or not there is a road to follow that will benefit the vehicle's travel through the corridor.

Path Planning takes all of the sensing and navigation information and decides on the appropriate behavior for reacting to the perceived situation. These decisions are then translated into a desired vehicle speed and a desired steering angle and sent to the Motion Control computer. This computer double checks the safety of the commands based on the current conditions of the vehicle. It sends a brake percent, throttle percent, and steering angle to the actuators in order to provide the desired motion. Figure 1 displays a block diagram of this processing architecture.

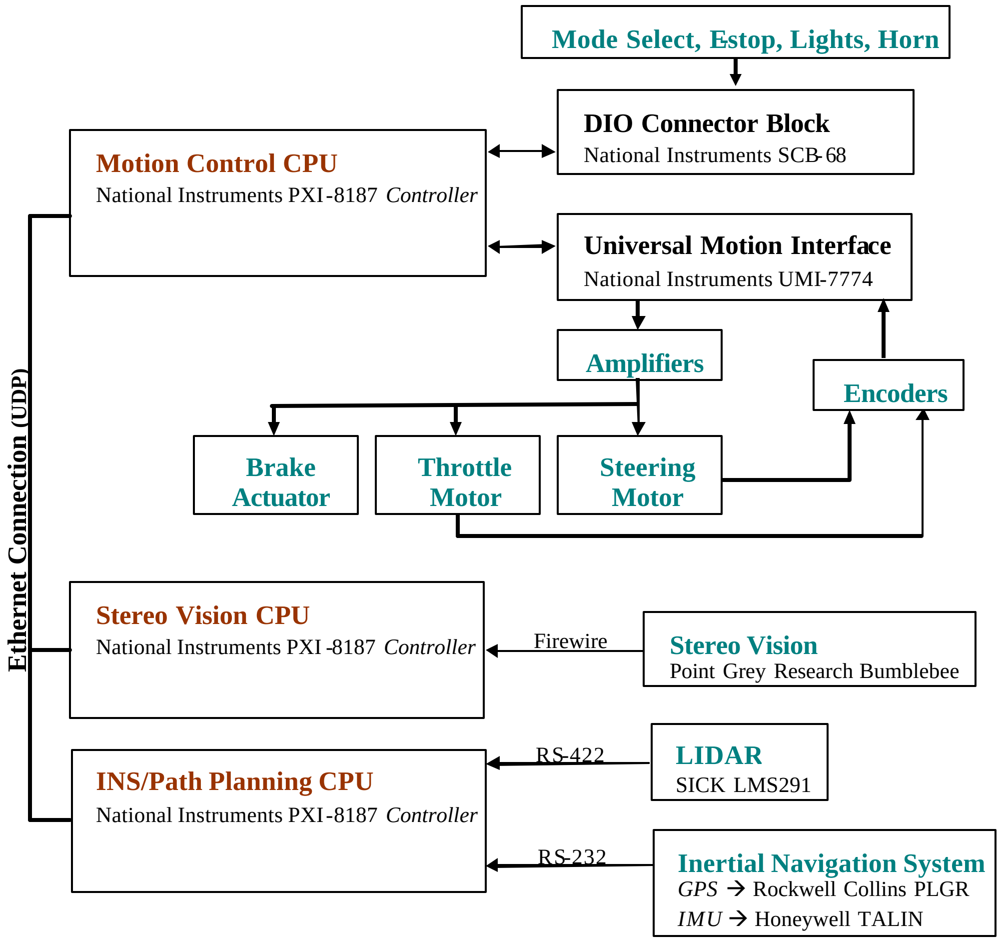

Figure 1. Block diagram of the processing architecture for the vehicle's sensing, navigation, and actuation

The computers on the vehicle are networked via an ethernet connection. Using UDP communication, the computers can send commands and sensor information to one another.

Figure 2 illustrates the network architecture and the information that is passed between each computer. The major challenge with the vehicle network has been handling communication failure. Since the main source of speed feedback comes from the INS, it is extremely important to make sure Cliff is receiving the correct data. UDP communication is used with a high data rate to pass information between computing components.

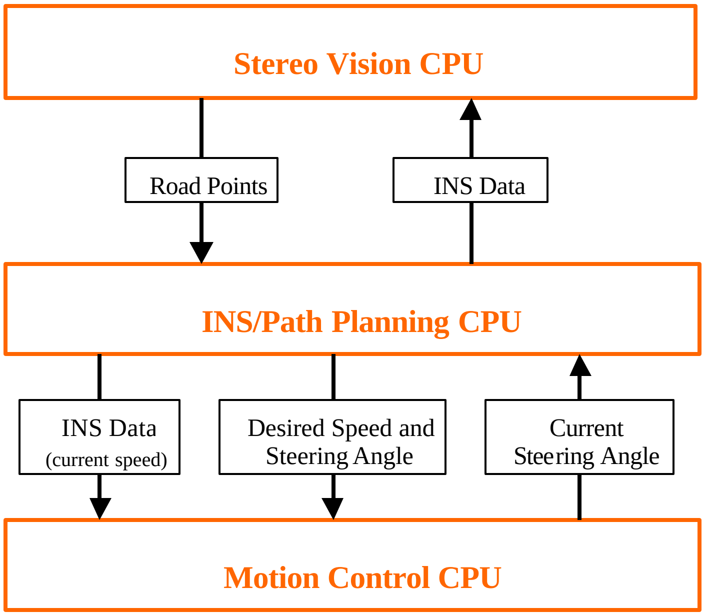

Figure 2. Network architecture

2.1.3. Describe unique methods employed in the development process, including model-driven design or other methods used.

Cliff's software development began by building a simulator with which to test our navigation strategies without taking the vehicle into the field. This has proven invaluable to the team in test time, early error detection, and software development time.

Comment [JSP1]: Why not use what we wrote in Rocky's?

2.2. Localization

2.2.1. Explain the GPS system used and any inertial navigation systems employed during GPS outages (as in tunnels). Include a discussion of component errors and their effect on system performance.

Cliff uses a Honeywell TALIN inertial navigation system coupled with a Rockwell-Collins PLGR GPS to determine position, velocity and attitude. With continuous updates from the GPS, the TALIN converges to an accurate (20cm, est.) solution within 15 minutes. This accuracy is maintained as long as the GPS continues to update. In the event of a GPS outage, the TALIN automatically switches to inertial navigation mode. Excluding an extended GPS outage (>20min in motion), we expect to see continuous accurate position information throughout the course.

2.2.2. If map data was an integral part of the vehicles navigation system, describe the requirements for this data and the way in which it was used.

No map data was used in Cliff's navigation system.

2.3. Sensing

2.3.1. Describe the location and mounting of the sensors mounted on the vehicle. Include a discussion of sensor range and field of view. Discuss any unique methods used to compensate for conditions such as vibration, light level, rain, or dust.

Cliff's sensors are arranged to maximize field of view and sensor range. The horizontally mounted scanning LIDAR scans a 180 degree arc in front of the vehicle and is hard-mounted in position. This sensor is used for primary obstacle detection. The advertised range of the LMS-291 is 80m, but Cliff's software only uses LIDAR data at a maximum range of 40m.

Comment [JSP2]: The horizontally hard-mounted scanning LIDAR scans a 180 degree arc in front of the vehicle.

The GPS antenna is mounted to the highest point on the vehicle roll cage to ensure maximum view of the sky.

The Point Grey Bumblebee stereo vision camera is mounted in the high-center-front of our roll cage to protect the sensor and maximize the view down onto the terrain ahead. The Bumblebee is capable of processing image points to a range of 30-40m and is used to find preferred areas of travel on the course. The team has implemented it's own software to compensate for varying light levels on the camera and exclude the sky data from the processed image.

Comment [JSP3]: Pixels beyond the 30-40m range (Ray told me this)

2.3.2. Discuss the overall sensing architecture, including any fusion algorithms or other means employed to build models of the external environment.

Cliff uses a hierarchical structure to fuse behaviors of three types: waypoint seeking, road following, and obstacle avoidance. This fusion is discussed more thoroughly in section 2.4.1. Data from the TALIN INS is used to compute the desired steering angle for the waypoint seeking behavior. If the bumblebee stereo vision camera detects a preferred path (road/dirt track) and the perceived path does not cause the vehicle to leave the RDDF-defined corridor, the road following behavior assumes control of the vehicle steering. If an obstacle is detected by the SICK LMS-291 scanning LIDAR, the obstacle avoidance behavior assumes control over vehicle steering and speed and remains in control until the potential collision has passed.

Comment [JSP4]: obstacle

2.3.3. Describe the internal sensing system and architecture used to sense the vehicle state.

Cliff uses an on-board accelerometer array with temperature sensor located in the electronics enclosure to measure the conditions to which the vehicle electronics are subject. Battery voltage is also logged on the vehicle's power system. All of these are recorded on the vehicle computers using a National Instruments USB DAQ. This information does not affect the vehicle's navigation behavior.

2.3.4. Describe the sensing-to-actuation system used for waypoint following, path finding, obstacle detection, and collision avoidance. Include a discussion of vehicle models in terms of braking, turning, and control of the accelerator.

During waypoint-only driving, Cliff perceives position, velocity, and attitude using the TALIN/PLGR INS. Cliff senses heading and applies a steering

Comment [JSP5]: Cliff uses the PLGR and TALIN in road following and obs avoid too angle to the wheels linearly proportional to the vehicle's deviation from the destination waypoint.

If a road is detected by the stereo vision cameras and does not cause the vehicle to deviate too far from the current waypoint, the vehicle is commanded to steer to the closest available point perceived to be a road centerline.

Cliff uses a reactive based obstacle avoidance strategy called Dynamic Expanding Zones (DEZ). This process overrides waypoint and road following behaviors. A SICK LMS-291 LIDAR scans for obstacles within the dynamically expanding avoidance zone. Once an obstacle or obstacles are identified, the vehicle will deviate from the waypoint/road point path to clear the avoidance zone of obstacles.

To accomplish all of these maneuvers, the path planning computer commands a speed and steering angle to the Motion Control PXI, which commands brake and throttle actuators for speed control and a DC servo motor for steering control. Custom-tuned PID controls are used for both speed and steering position control.

Comment [JSP6]: When a road is not detected and no obstacles exist, Cliff solely relies on position, velocity, and attitude to perform point-to-point waypoint navigation.

Comment [JSP7]: sounds wierd

2.4. Vehicle Control

2.4.1. Describe the methods employed for common autonomous operation contingencies such as missed-waypoint, vehicle-stuck, vehicle-outside-lateral-boundary-offset, or obstacle-detected-in-path.

Cliff uses a reactive based obstacle avoidance strategy called Dynamic Expanding Zones (DEZ). A SICK LMS-291 LIDAR scans for obstacles within the dynamically expanding avoidance zone. Once an obstacle or obstacles are identified, the vehicle will deviate from the waypoint/road point path to clear the avoidance zone of obstacles. Figure 2 displays the general principle behind DEZ obstacle avoidance.

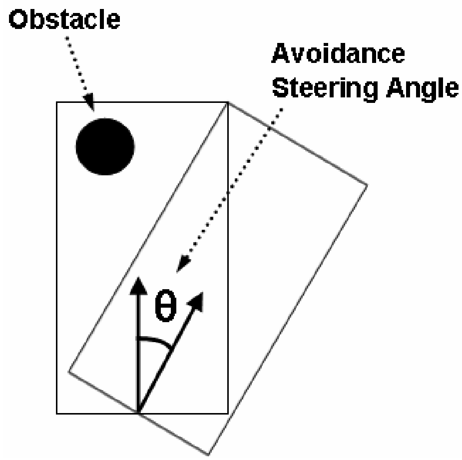

Figure 1. Illustration of how Box Avoidance determines a steering angle to clear the avoidance box.

After the avoidance zone is clear, the obstacle will enter the buffer zone, which is shown in Figure 2. The buffer prevents the vehicle from turning back into an obstacle. Once the obstacle has cleared the buffer, the vehicle will return to waypoint navigation and road following.

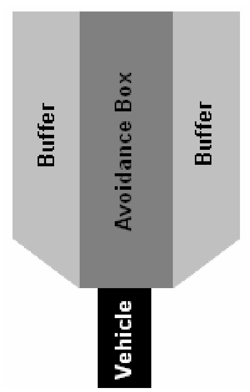

Figure 2. General layout of the avoidance box and buffers in relation to the vehicle.

All waypoint navigation is point to point, so the vehicle will always attempt to steer toward a waypoint if no obstacle is in the way. The vehicle will start to track the next waypoint once the vehicle enters the waypoint radius. The waypoint radius is dynamically controlled based on the angle between the waypoints. As the turning angle increases, the turn radius decreases. The minimum size of the waypoint radius is the smallest lateral boundary offset (LBO) around the waypoint. The waypoint radius will likely prevent the vehicle from skipping a waypoint unless the corridor is completely blocked or the vehicle exits the corridor.

Comment [JSP8]: it

Comment [JSP9]: waypoint

To prevent the vehicle from exiting the LBO, the boundary is considered an obstacle. In addition, a fuzzy logic controller is built into the obstacle avoidance to assist in keeping the vehicle inside the boundaries. In the event the vehicle does exit the boundary, it will return to the inside of the LBO immediately.

If vehicle becomes physically stuck, it will continue to increase the throttle to attempt free itself. Currently, no other measures exist to free the vehicle.

2.4.2. Describe the methods used for maneuvers such as braking, starting on a hill, or making a sharp turn without leaving the route boundaries.

Cliff has multiple levels of speed and steering control. The DEZ navigation program controls the first level speed and steering commands. For example, to optimize obstacle avoidance, the vehicle will reduce the speed based on the distance to an obstacle and angle to the waypoint. If the vehicle is executing a sharp turn, speed is controlled by the current steering angle, the difference between the current steering angle, and desired steering angle. Our speed control PIDs are capable of maintaining speeds ranging from 1mph to 25 mph. With the combination of excellent slow speed control and a tight turning radius Cliff should have no problems executing extremely sharp turns without exiting the route boundaries.

The Motion Control program receives these speed and steering commands and determines if they are safe from causing a rollover. In the event that the vehicle is likely to roll with the current/desired conditions, the Motion Control program will command a safe steering angle and speed.

Speed and steering are both controlled using PID control. So when starting on a hill, the PID will increase the throttle percent until the vehicle speed tracks the desired speed.

Brake percent is controlled by the current steering angle, roll angle, amount of speed reduction, and urgency. These controls prevent a rollover from occurring and prevents the vehicle from slamming on the brakes when its not urgent to slow quickly.

2.4.3. Describe the method for integration of navigation information and sensing information.

Obstacle avoidance has precedence over strict waypoint navigation and road following. If the LIDAR recognizes the boundary or an obstacle in the way of the vehicle path, the vehicle will steer to avoid it. However, the steering fuzzy logic controller prevents obstacle avoidance from making the vehicle completely ignore the waypoint and driving off the course.

Once the vehicle has successfully avoided the obstacle, the vehicle will return to following the waypoint or road point. Road center points are provided by the Stereo Vision camera. Any points outside of the corridor are automatically discarded. Fuzzy logic control determines whether the road points in the corridor follow the general direction of the waypoint.

2.4.4. Discuss the control of the vehicle when it is not in autonomous mode.

The onboard joystick is used to control the vehicle when it is not in autonomous mode. Cliff's joystick is input using two of the analog channels on the universal motion interface board. This allows the driver to command a desired steering angle and throttle percent. These values are then sent to the actuators just as if the autonomous controls had commanded it.

The vehicle also has two pedals that control two different brake systems. There is a control brake pedal that, like the joystick, is an analog input that ties into the computer control system. The driver can use this pedal to control the brake percent that is commanded to the brake actuator. No matter what mode the vehicle is in, the control brake pedal has the ability to override the commanded brake percent. The second pedal is the manual brake that uses the parking brake system that came with the vehicle. This provides a way for the driver to stop the vehicle from within if there is a computer failure.

Cliff has five modes: (1) Stop, (2) Joystick, (3) Computer Steering/Joystick Throttle, (4) Computer Throttle/Joystick Steering, and (5) Fully Autonomous. Stop mode simply commands zero to the steering, brake and throttle actuators. Joystick mode gives the driver full control of all three of the actuators. The next two modes are used mainly for testing and debugging purposes. Computer Steering/Joystick Throttle provides a way to test navigation without having to worry about the computer speed (throttle or brake). Computer Throttle/Joystick Steering provides a way to test speed control. Fully autonomous mode gives the control of all three of the actuators to the computers.

2.5. System Tests

2.5.1. Describe the testing strategy to ensure vehicle readiness for DGC, including a discussion of component reliability, and any efforts made to simulate the DGC environment.

Cliff was subjected to extensive simulated and live testing in preparation for the 2005 Grand Challenge. Most live testing took place in an approximately 75 acre field in Blacksburg, Virginia. This area allowed for various courses to be set up where Cliff was exposed to hills, rough terrain, and dirt roads. Numerous long-term tests were also conducted in order to test the mechanical, electrical, and software reliability. In these tests, Cliff was run in autonomous mode over a looped course until an error caused it to stop. By running these tests, long term software and hardware reliability could be determined, and changes could be made to allow the vehicle to run for longer periods of time.

A vehicle simulator program was also designed to test conditions and situations that would be difficult, if not impossible, for Cliff to encounter in

Blacksburg. This program creates a virtual map and sensor data that is relayed to the actual pieces of software that control the vehicle. This simulator, along with information about Cliff's vehicle dynamics, tested the algorithms in a virtual space before ever placing them on the vehicle. It also allowed for testing during conditions where it would normally not be possible, such as at night or times when In order to attempt and simulate the desert environment that will be seen at the Grand Challenge, data was taken from deserts in Arizona and Texas. This data was used in correlation with the simulator to determine how Cliff would react to heavy dust clouds, rocky hills, and various other desert conditions.

A second set of software allowed various data recorded from the vehicle to be replayed for analysis. This replay software played back information such as vehicle position and orientation, speed, throttle and brake percentages, and LIDAR scans at the same speed that it was originally recorded. Being able to play back exactly what happened during autonomous runs is valuable to determine exactly how Cliff behaved in the real world.

2.5.2. Discuss test results and key challenges discovered.

From the numerous tests that Cliff underwent, a number of key tests resulted in changes to the software and hardware. The main issued that appeared early on during testing was the recognition of objects that would not be considered obstacles. Certain objects, such as tall grass or hills, could be seen by the LIDAR scans as variations to normally flat terrain, but would not be considered obstacles for the vehicle to pass. In order to handle these situations various parts of the software, such as the zones where obstacles are detected, were changed in order to account for these fake obstacles. Another problem arose where Cliff would avoid an obstacle, but attempt to turn back towards the center of the corridor before the obstacle was clear. To fix this problem, while avoiding obstacles the detection zones are expanded laterally from the vehicle. This allowed Cliff to know when an obstacle had been passed, and it was safe to return to the center of the corridor.

Comment [JSP10]: lateral expanding buffer zones were added to prevent the vehicle from turning into an obstacle

Other than numerous minor vehicle failures, the biggest problem that arose was a lack of power from the onboard batteries. During long term testing sessions, suitable power was not being supplied to battery-controlled devices, such as the steering and throttle motors. It was discovered that the chargers used to re-supply the batteries with power during operation were not charging as fast as other systems were drawing power away. New chargers were purchased in order to ensure all devices on the vehicle receive adequate power at all times.